Neil Rolf

CSC300

Program 7 – Design

**Class Declarations**

typedef std::string SElement

Class HashPython

*private variables*

* struct TableElem
  + SElement element; //holds string element
  + unsigned count; //holds number of string element instances
* enum TableLimit {TABLE\_SIZE = 41} //sets size of hash table
* TableElem hashTable[TABLE\_SIZE] //defines the hash table
* SElement userFile //defines the file name to read
* int getHashKey(SElement) //calculates hash key, uses quadratic collision avoidance
* void ingest() //reads string elements from file to hash table, calls getHashKey and findWord for each string element

*public variables*

* HashPython() //default constructor, calls clearCounts & ingest
* HashPthon(SElement) //parameterized constructor, accepts file name, calls clearCounts & ingest
* void clearCounts() //sets element and count members to 0 or null
* bool findWord(const SElement&); //searches the hash table for a word and increments the count member for each instance
* void printHashTable() //prints contents of hash table

**Driver**

enum Menu {ENTER\_FILENAME=1, EXIT};

int(main){

//USER MENU

int selection = 0;

do{

cout << “user menu and selection prompt”;

cin.clear();

cin >> selection;

switch(selection){

case ENTER\_FILENAME:{

cout << “prompt user for file to read”;

cin >> string file;

HashPython list(file); //class init, calls parameterized constructor

break;}

case EXIT:{

break;}

}while(selection !=2);

return 0;

}